



ARL-MR-0944 • OCT 2016



Implementation of Unsteady Double-Axis of Rotation Motion to Predict Pitch-Damping Moment

by Joseph D Vasile

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Implementation of Unsteady Double-Axis of Rotation Motion to Predict Pitch-Damping Moment

by Joseph D Vasile

Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) October 2016		2. REPORT TYPE Memorandum Report		3. DATES COVERED (From - To) January 2016–March 2016	
4. TITLE AND SUBTITLE Implementation of Unsteady Double-Axis of Rotation Motion to Predict Pitch-Damping Moment				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Joseph D Vasile				5d. PROJECT NUMBER AH80	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WML-E Aberdeen Proving Ground, MD 21005-5069				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-MR-0944	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>An investigation was performed to verify the implementation of double-axis rotation of a well-studied projectile (Army-Navy Finner) using an existing commercial fluid-flow solver by computing the pitch-damping moment (PDM) in multiple manners and comparing to previously published results. The PDM stability derivative was computed using 3 methods: 1) planar pitching, 2) lunar coning (steady and transient), and 3) equal and opposite coning-spinning motion. The investigation studied the effect of time step for the transient double-axis rotation simulation. Overall, the predicted body moments agreed quite well with results, suggesting that convergence can be achieved with further reduced global time step size. Although the unsteady double-axis rotation implementation study proved to be successful (i.e., verifying that the method can be used in transient simulations of projectiles undergoing complex kinematic motions), it is inefficient, and, therefore, may not be an advisable method to accurately predict PDM.</p>					
15. SUBJECT TERMS <p>CFD++, pitch-damping moment, PDM, computational fluid dynamics, double-axis rotation, CFD, coning, spinning, 6-DOF, lunar coning, planar pitching, Army-Navy Finner</p>					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 34	19a. NAME OF RESPONSIBLE PERSON Joseph D Vasile
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-306-1794

Contents

List of Figures	v
List of Tables	v
Acknowledgments	vi
1. Introduction	1
2. Pitch-Damping Moment (PDM) Methodology	2
2.1 Planar Pitching	2
2.2 Lunar Coning	3
2.3 Coning-Spinning	4
3. Computational Setup	6
3.1 Computational Domains and Boundary Conditions	6
3.2 Numerics	6
4. Implementation of Double-Axis Rotation	7
4.1 Body Spinning Motion	9
4.2 Coning Rotation Motion	11
4.3 Kinematic Check	14
4.4 Output	14
5. Results	17
6. Conclusions	20
7. References	22
Appendix. Transformation from Inertial (I) to Nonrolling Coning Frame (B)	23

List of Symbols, Abbreviations, and Acronyms	25
Distribution List	26

List of Figures

Fig. 1	Schematic of Army-Navy Basic Finner (ANF), 1 caliber = 0.03 m	2
Fig. 2	Cm as a function of alpha for transient planar pitching.....	3
Fig. 3	Schematic of coning and body spinning motion.....	4
Fig. 4	Example of lunar coning motion (no body spin) (video).....	5
Fig. 5	Example of double-axis rotation (coning with counter-rotation body spin) (video)	5
Fig. 6	ANF computational domain rotated to $\alpha = 0.5^\circ$ about the center of gravity of the ANF geometry	8
Fig. 7	Grid motion/coordinate systems menu	9
Fig. 8	Define body spinning rotation	10
Fig. 9	File-based body spinning rotation motion	11
Fig. 10	The 6-DOF menu	12
Fig. 11	The 6-DOF specifications	12
Fig. 12	Definition of body boundaries	13
Fig. 13	Kinematic motion of double-axis rotation (coning with counter-rotation body spin) (video).....	14
Fig. 14	Example output of mcfd6dof_b1.dat	15
Fig. 15	Cm in inertial and body frame $0.25^\circ \Delta t$	16
Fig. 16	Cn in inertial and body frame $0.25^\circ \Delta t$	16
Fig. 17	Cl in inertial and body frame $0.25^\circ \Delta t$	16
Fig. 18	Transformation of forces and moments from inertial to body frame for transient lunar coning motion $0.5^\circ \Delta t$	19

List of Tables

Table 1	Comparison of steady and transient lunar coning.....	18
Table 2	Comparison of transient methods computing PDM	19
Table 3	Comparison of computed PDM	20

Acknowledgments

The author would like to thank Mr Vishal Bhagwandin for providing both the geometry and computational domain of the Army-Navy Finner and Drs Sidra Siltan, Jim DeSpirito, and Paul Weinacht for guidance, mentoring, and many discussions relating to both projectile flight dynamics and computational fluid dynamics computations. This work was supported in part by a grant of high-performance computing time from the US Department of Defense (DOD) High-Performance Computing Modernization program at the US Army Research Laboratory DOD Supercomputing Resource Center, Aberdeen Proving Ground, Maryland.

1. Introduction

The flight dynamics of a projectile can be quite complex, usually comprised of body motion that undergoes rotations about multiple axes. To accurately predict the aerodynamic effects of the projectile during flight, the motions of the projectile must be accurately modelled. Typically, when computing aerodynamic coefficients, motion about each axis is considered individually (i.e., spin around body-axis, pitch about side-axis). There may be situations that arise that require the aerodynamic coefficients to be determined when motion is occurring about more than one axis at a time, such as spin about the body axis and coning about the velocity vector. It is important to ensure that such a motion is implemented and interpreted correctly.

To this end, the current investigation implements a double-axis rotation on a well-studied projectile to determine if the correct aerodynamic forces and static and dynamic moments can be accurately predicted while undergoing the prescribed motion. Following the approach of Weinacht et al.,¹ a combination of spinning and coning motion was implemented to determine the pitch-damping moment (PDM) coefficient using the body side moment coefficient of a projectile. The intent of the study is solely to verify the implementation of double-axis rotation method, and not necessarily advisable to be used in predicting PDM. By applying this procedure to obtain known PDM coefficient results, this investigation permits greater understanding in defining the motion of projectiles undergoing prescribed double-axis rotation within CFD++ (version 15.1.1), a commercial fluid-flow solver developed by Metacomp Technologies, Inc.²

The Army-Navy Finner (ANF) geometry was used to validate the double-axis of rotation motion implementation by comparing the results from the current simulations to that of the detailed work of Bhagwandin and Sahu.³ The ANF geometry has a diameter, d , of 0.03 m (1 caliber) and consists of a 10° half-angle cone that is 2.84-calibers long, followed by a 7.16-caliber cylindrical body (Fig. 1). There are four 1- × 1-caliber fins with sharp leading edges and thicknesses of 0.08 calibers at the trailing edge.

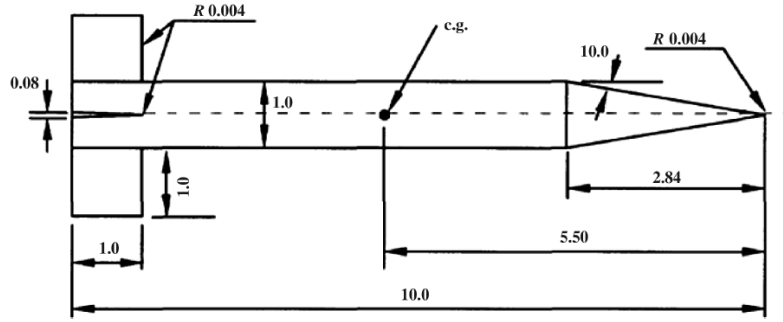


Fig. 1 Schematic of Army-Navy Basic Finner (ANF),⁴ 1 caliber = 0.03 m

2. Pitch-Damping Moment (PDM) Methodology

The PDM is a dynamic stability derivative that is used in stability analyses to determine if a projectile is stable during flight. In this investigation, 3 methods were implemented in CFD++ to determine the pitch damping motion for the ANF geometry at a single specified Mach number ($M = 2.5$): transient planar pitching, lunar coning (steady-state and transient), and transient spinning-coning.

2.1 Planar Pitching

As described in detail in Bhagwandin and Sahu³ and DeSpirito et al.,⁵ the planar-pitching method directly solves for the PDM. Transient planar pitching is the motion whereby the projectile harmonically oscillates about its center of gravity in rectilinear flight. Following the procedure of Bhagwandin and Sahu,³ time-accurate computational fluid dynamics (CFD) methods are used to compute the time dependent, forced, sinusoidal motion flow solution. The forced oscillation produces a hysteresis variation of pitching moment with α , symmetric about α_0 (the mean angle of attack). An example of the hysteresis of C_m is shown in Fig. 2. Therefore, the PDM can be computed using the 2 pitching-moment values (pitch up and pitch down) where the projectile passes through α_0 . The PDM for transient planar pitching can be computed as follows:

$$PDM = \frac{C_{m_u} - C_{m_d}}{2kA}, \quad (1)$$

where C_{m_u} , C_{m_d} , and A , is the pitching moment value during pitch-up and pitch-down motions, and amplitude of motion, respectively. Moreover, the reduced pitch frequency, k , is defined as

$$k = \frac{\omega_r d}{2V_\infty}, \quad (2)$$

where ω_r is the frequency of the oscillation in rad/sec and V_∞ is the freestream velocity. The reduced frequency and amplitude of the oscillation was selected to match values that were used in Bhagwandin and Sahu,³ ($k = 0.05$ and $A = 0.25^\circ$, respectively).

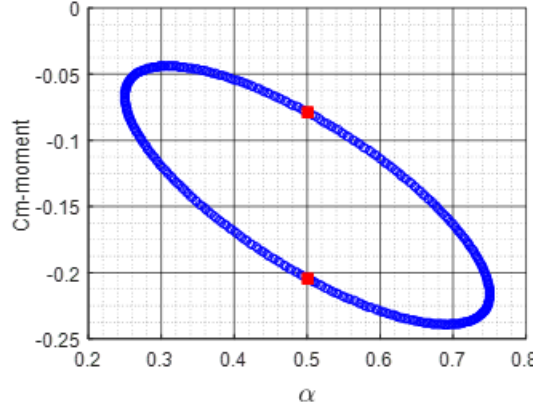


Fig. 2 Cm as a function of alpha for transient planar pitching

2.2 Lunar Coning

Lunar coning is the motion whereby the projectile flies at a constant angle, α , with respect to the freestream velocity vector while undergoing a constant angular rotation, Ω , about a line coincident to the freestream velocity vector passing through the center of gravity of the projectile. For lunar coning motion, the body side moment, C_n , is related to the PDM as follows:

$$PDM = \frac{C_n}{\sin(\alpha) \cdot \hat{\Omega}} - C_{n_{p\alpha}}, \quad (3)$$

where $\hat{\Omega}$ is the nondimensional angular coning rate and $C_{n_{p\alpha}}$ is the Magnus moment derivative. The nondimensional angular coning rate, $\hat{\Omega}$, is defined as the following:

$$\hat{\Omega} = \frac{\Omega d}{2V_\infty}. \quad (4)$$

The nondimensional angular rate and coning angle were selected to match the values chosen from the work of Bhagwandin and Sahu³ ($\hat{\Omega} = 0.0025$, $\alpha = 0.5^\circ$). These values ensured that the linear assumptions of lunar coning motion were met. The lunar coning motion is a one-axis rotation about the velocity vector (coning) and is typically implemented through a steady-state simulation.^{3,5,6} For the work presented here, in addition to the steady-state simulation, a transient simulation was completed using the prescribed double-axis motion for comparison (with zero-body spin motion).

The lunar coning motion imparts a relative spin to the projectile about its longitudinal axis, producing a small Magnus component, which must be accounted for when determining the PDM sum. This can be done in one of 2 ways. The first option is to complete a separate calculation to determine the Magnus moment derivative. For non-axisymmetric projectiles, such as the ANF, the Magnus moment derivative must be calculated from a separate transient simulation, where the body undergoes a spinning motion about its body axis, at a constant angle, α , with respect to the freestream. For this investigation, the Magnus moment derivative was not calculated, and therefore only the contribution of coning motion to PDM was compared. The second option is to have the body spin to oppose the coning rate. In the following section of the report, the prescribed motion of coning and body spinning equal and opposite in direction in order to correct for this present Magnus component is discussed.

2.3 Coning-Spinning

The third method to calculate PDM was to determine the body side moment for the projectile as it undergoes a coning motion and a body spin (double-axis) motion that is approximately equal and opposite of the spin induced by the coning motion (Fig. 3). The PDM can then be computed using Eq. 5. This method is the same as that described by Weinacht et al.¹ For the current study, a transient, double-axis simulation was implemented in CFD++.

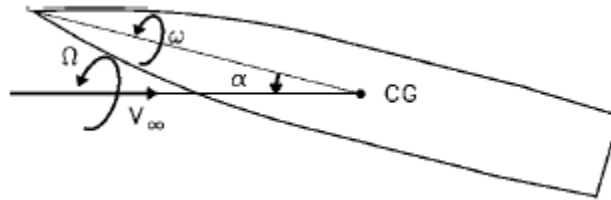


Fig. 3 Schematic of coning and body spinning motion¹

$$PDM = \frac{C_n}{\sin(\alpha) \cdot \Omega} \cdot \quad (5)$$

The coning motion, Ω , is a rotation of the projectile's longitudinal axis about the freestream velocity vector, and the body spinning motion, ω , is a rotation of the projectile about its longitudinal axis. Following the above relationship, the body spinning rotation is equal in magnitude but opposite in sign to the component of Ω along the longitudinal axis (α is the initial coning angle). This relationship can be explained mathematically:

$$\Omega = \frac{\hat{\Omega} \cdot 2 \cdot V_{\infty}}{d}, \quad (6)$$

$$\omega = -\Omega \cos(\alpha). \quad (7)$$

The nondimensional angular coning rate ($\hat{\Omega} = 0.0025$) was selected to match the lunar coning rate used earlier. For this investigation, an initial freestream velocity of 851 m/s ($M = 2.5$) and coning angle of 0.5° were used in order to compare forces and moments to the findings of Bhagwandin and Sahu,³ resulting to $\Omega = 141.83$ rad/s and $\omega = -141.82$ rad/s, respectively.

The kinematic motions between lunar coning (no prescribed body spin) and double-axis rotation (coning with counter-rotation body spin) are shown in Figs. 4 and 5, respectively). Lunar coning is the unique double-axis motion of zero-body spin while undergoing coning. The lunar coning motion imparts a rotation on the projectile in the body reference frame. When the projectile undergoes combined coning-spinning motion, the total angular velocity of the body about the longitudinal axis is zero.

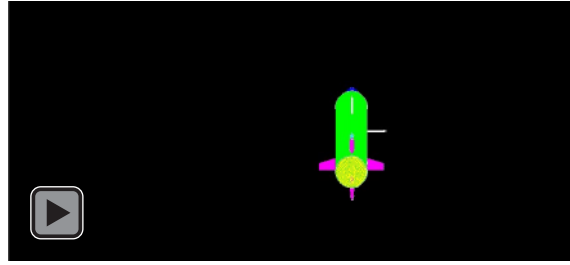


Fig. 4 Example of lunar coning motion (no body spin) (video)

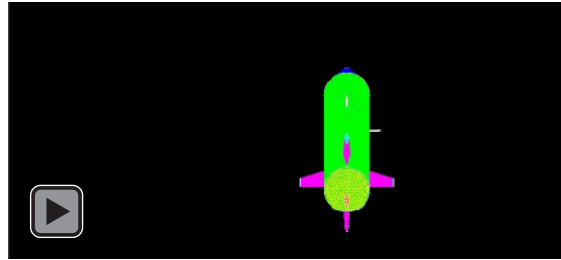


Fig. 5 Example of double-axis rotation (coning with counter-rotation body spin) (video)

3. Computational Setup

3.1 Computational Domains and Boundary Conditions

The same computational grid from Bhagwandin and Sahu³ for the ANF geometry was used for all simulations. The mesh was created using Pointwise V16.03R4, comprised of 17M structured hexahedral cells. The domain extended approximately 60 calibers in the radial direction to form a spherical farfield boundary, where a characteristics-based inflow/outflow boundary condition was applied. The projectile walls were modeled as viscous adiabatic walls using a solve-to-the-wall strategy where the initial grid spacing normal to the walls was 0.0005 mm ($y^+ < 1$). The grid was decomposed on 120 processors for parallel computation. The computations were performed on “Excalibur”, a Cray XC40 supercomputer consisting of 3,098 compute nodes, 32 Intel Xeon E5-2698 v3 cores per node, 128-GB memory per node and a Cray Aries interconnect. Excalibur is housed and managed by the US Army Research Laboratory (ARL) Department of Defense Supercomputing Resource Center (ARL DSRC) at Aberdeen Proving Ground, Maryland.

3.2 Numerics

CFD++ version 15.1.1 is a commercial CFD finite volume, unstructured solver by Metacomp Technologies, Inc. The 3-dimensional compressible Reynolds Averaged Navier-Stokes (RANS) equations are solved using a finite volume method with a point-implicit time integration scheme for advancing the solution in both steady and transient simulations. Although many turbulence closure models are available within CFD++, all simulations were completed using the 2-equation, realizable k - ϵ model. Both steady-state and transient simulations were employed.

For all steady-state simulations, a linear ramping schedule was used to gradually increase the Courant-Friedrichs-Lewy (CFL) number from 1 to 30 over the first one hundred iterations, after which a constant CFL of 30 was maintained. A 1-order to 2-orders of magnitude blending spatial discretization method was implemented, where the solution is started at 1 order of magnitude to aid convergence, then transitions to 2 orders of magnitude over 50 iterations and then remains fully 2 orders. For this study, the 1–2-orders of magnitude transition ended at 250 iterations. Convergence for the total forces and moments was typically achieved in a few thousand iterations, with residuals reducing at 5 orders or more in magnitude. All forces and moments were averaged over the final 200 iterations.

For all transient RANS simulations, dual-time stepping was used, where a global (physical) time step is specified and an inner (pseudo) time step is determined via the maximum CFL number of the steady-state simulation. For the transient planar pitching simulations, the global time step of the transient simulation was selected based on 200 steps per pitch oscillation cycle. A total of 3 pitch cycles were simulated, with 20 inner iterations per global time step. For the double-axis rotation simulations, the global time step was studied using 360, 720, and 1,440 steps per coning cycle. Four coning cycles were simulated at each time step, with 20 inner iterations per global time step. All forces and moments were averaged over the final 2 coning or pitching cycles.

The intent of the current report is to guide users on how to implement a complex kinematic motion in CFD++. The next section describes in detail the procedure to implement a double-axis rotation, coning and body spinning, CFD simulation.

4. Implementation of Double-Axis Rotation

The double-axis rotation was implemented in CFD++ by specifying 2 simultaneous body-frame grid velocities. To the best of the author's knowledge, the double-axis rotation motion can only be performed for transient simulations in CFD++ due to its dependence on utilizing the 6-degree-of-freedom (DOF) specification frame option to simulate the prescribed motion. In this example, a coning-spinning rotation motion was implemented. Excerpts of this section were taken from a 2011 Metacomp Technologies, Inc. tutorial titled, "Double-Axis Rotational Body Motion in CFD++".⁷

In the present example, the global +x axis is along the streamwise direction (pointing downstream), global +z axis is along the cross stream direction (pointing upward), and global +y axis pointing into the page (Fig. 6). The global axis is the coordinate system defined by the initial mesh and geometry loaded into CFD++, and therefore becomes important when eventually defining initial Euler angles. This coordinate system defines the inertial reference frame.

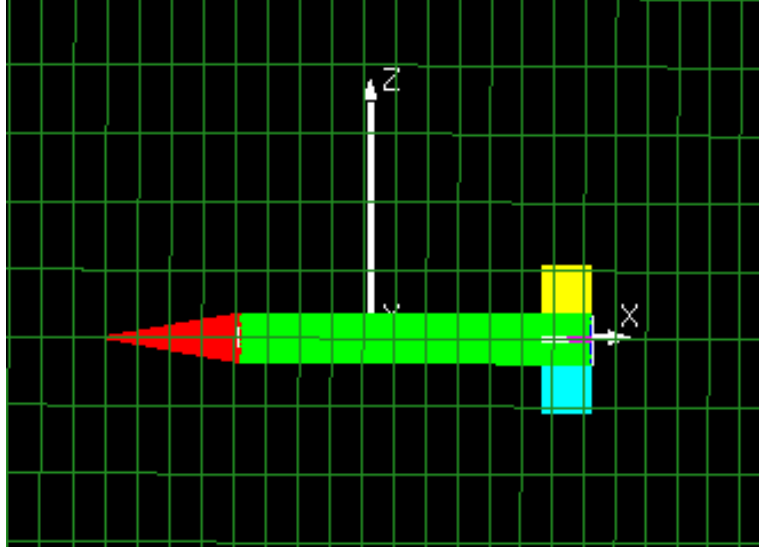


Fig. 6 ANF computational domain rotated to $\alpha = 0.5^\circ$ about the center of gravity of the ANF geometry

For this example, the ANF geometry was simulated with a coning angle (α) of 0.5° with respect to the freestream (Fig. 6). A steady-state solution is used to initialize the transient solution. To obtain the steady-state solution, the mesh was rotated $+0.5^\circ$ about the y-axis such that the body axis is now inclined to the freestream flow, which remains aligned with the global +x axis.

To prescribe the double-axis rotation, the body-frame grid velocity options are used. In the CFD++ GUI, these can be accessed by selecting “Topology >> Grid Motion/Coordinate Systems >> Special Body Frame” options. The rotation about the body axis (spinning motion) is setup using the “File-Based Single-Axis Rotational motion” option, and the rotation about the velocity vector (coning motion) is specified in the “Six-DOF Body Specification” (Fig. 7). This ensures that the force and moment output is in the body frame.

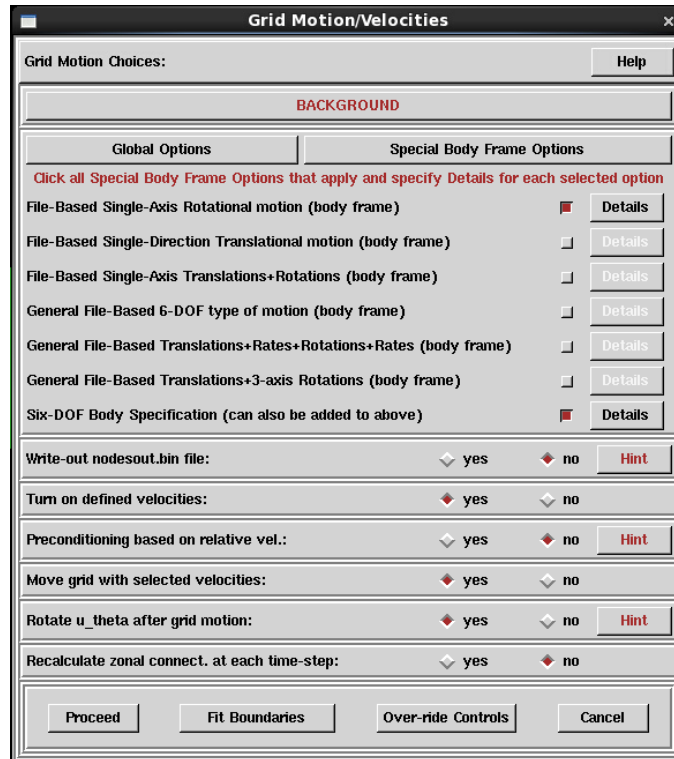


Fig. 7 Grid motion/coordinate systems menu

In the Grid Motion/Velocity information panel that follows, the user should select “yes” to turn on grid velocities and to allow the grid to move.

4.1 Body Spinning Motion

After clicking on “Details” for “File-Based Single-Axis Rotational motion (body frame)”, the user is prompted for the number of sets of grid velocity specifications. For the present case of a single geometry, “1” should be entered.

The cell group number to which the velocities will be applied must be specified in the first entry box. If the velocities should be applied to all groups, “0” must be entered. If there are multiple cell groups, and the velocities are to be applied to only one group, then the corresponding group number should be entered. A data file must be created to govern the rotational motion of the body. The name of the data file must be entered in the second entry box. The format of this file is described in the red text in the “Grid Velocity Information” panel (Fig. 8). Click “Accept and Exit”.

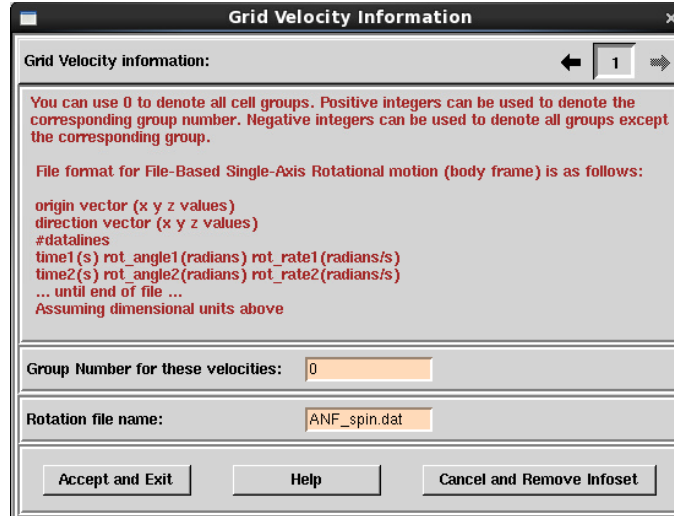


Fig. 8 Define body spinning rotation

The rotation data file, ANF_spin.dat, was used in this example as shown in Fig. 9. The first line specifies the location of the origin of the axis of rotation. This was chosen as the center of gravity of the projectile, which happens to be coincident with the global origin. The second line specifies the direction vector of the axis of rotation (with respect to body). The spinning motion was selected to be along the longitudinal axis of the projectile (body frame). The third line gives the number of lines of data that will be used to define the rotational motion. Each line of data specifies the global time (s), rotation angle (radians), and rotational rate (rad/s). One needs to provide at a minimum 2 global times (s), the corresponding rotation angles (radians), and the desired rotation rate (rad/s). CFD++ linearly interpolates the rotation angle from this information. In the rotation data file, the rotation rate is independent of the time and rotation angle. Thus, for a case such as this, where the rotation rate is constant, it should be specified as the same values on all lines. In this example, the rotation rate is $\omega = -\Omega \cdot \cos(\alpha)$, (spin motion in rad/s), at global time $t = 0$ s and at an arbitrary global time $t = 6$ s. The rotation angle at 6 s was computed as $\omega \cdot t$, because ω is constant. Click “Accept and Exit”.

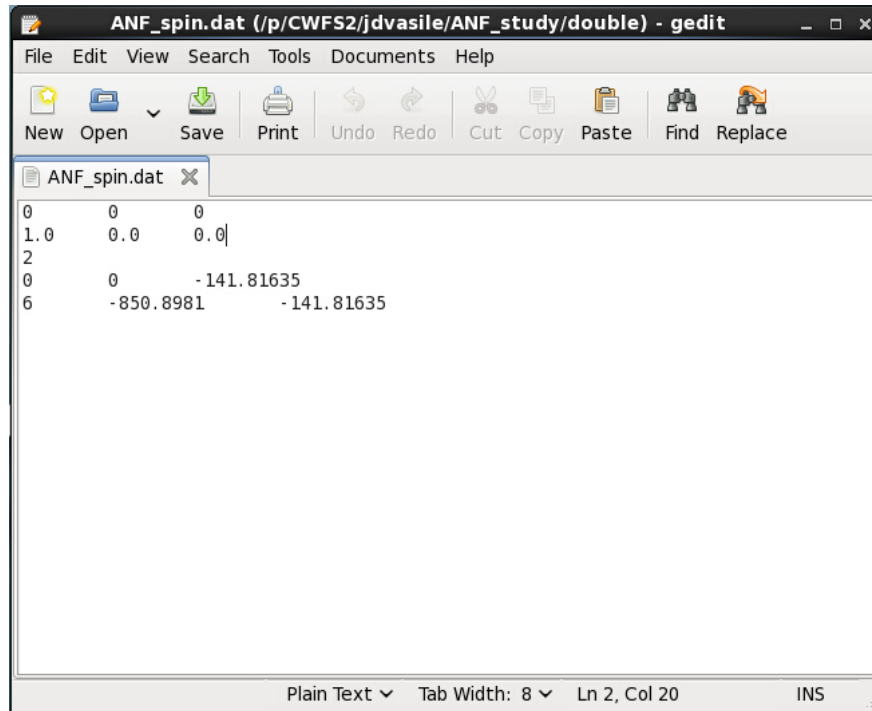
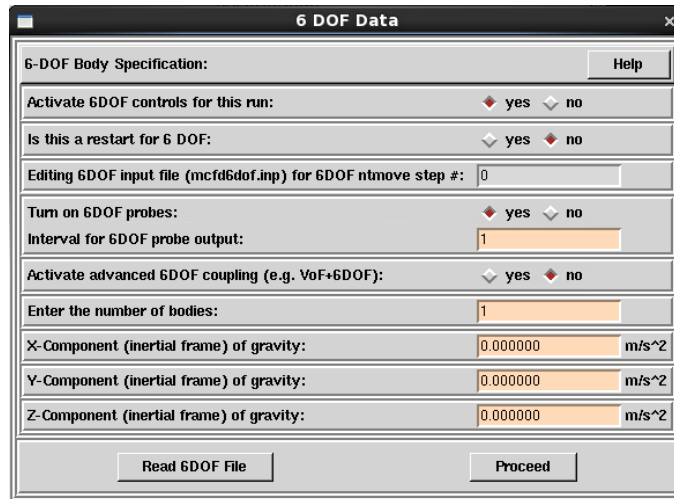


Fig. 9 File-based body spinning rotation motion

4.2 Coning Rotation Motion

The rotation of the body axes about the global x-axis can be specified by the 6-DOF Body Specification. After clicking on “Details”, the user must input some basic options in the first 6-DOF panel. The user must select “yes” to activate the 6-DOF controls and to move the grid with 6-DOF generated velocities. The number of bodies in motion must be input in the appropriate entry box, and all gravitational components should be set to 0 since the body motion is being directly prescribed—not computed based on external forces (Fig. 10). Click “Proceed”.



6 DOF Data

6-DOF Body Specification: Help

Activate 6DOF controls for this run: ☒ yes ☐ no

Is this a restart for 6 DOF: ☐ yes ☒ no

Editing 6DOF input file (mcf6dof.inp) for 6DOF ntmove step #:

Turn on 6DOF probes: ☒ yes ☐ no

Interval for 6DOF probe output:

Activate advanced 6DOF coupling (e.g. VoF+6DOF): ☐ yes ☒ no

Enter the number of bodies:

X-Component (inertial frame) of gravity: m/s²

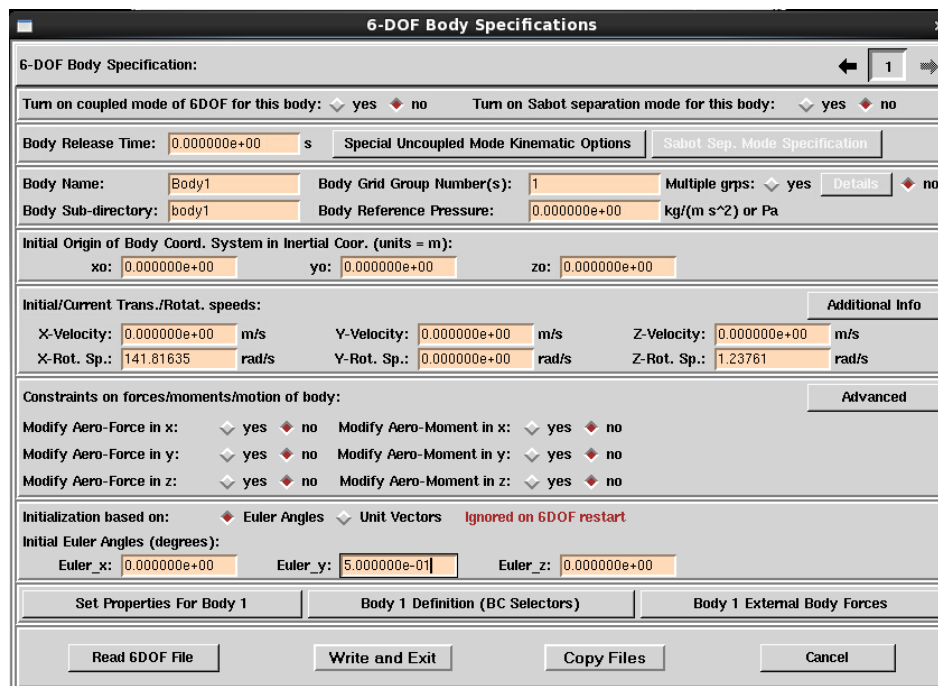
Y-Component (inertial frame) of gravity: m/s²

Z-Component (inertial frame) of gravity: m/s²

Read 6DOF File Proceed

Fig. 10 The 6-DOF menu

Figure 11 presents the 6-DOF specifications. Select “no” for “Turn on coupled mode of 6DOF for this body:”. This choice allows the velocities to be prescribed directly rather than being computed based on external forces. The body grid group number must be entered in the appropriate box. If the user wishes to shift the initial position of the body grid, this can be done by modifying the “Initial Origin of Body Coord. System in Inertial Coord. (units = m):”. In this example, no shift was imposed.



6-DOF Body Specifications

6-DOF Body Specification: ← →

Turn on coupled mode of 6DOF for this body: ☐ yes ☒ no Turn on Sabot separation mode for this body: ☐ yes ☒ no

Body Release Time: s Special Uncoupled Mode Kinematic Options Sabot Sep. Mode Specification

Body Name: Body Grid Group Number(s): Multiple grps: ☐ yes Details ☒ no

Body Sub-directory: Body Reference Pressure: kg/(m s²) or Pa

Initial Origin of Body Coord. System in Inertial Coord. (units = m):
 xo: yo: zo:

Initial/Current Trans./Rotat. speeds: Additional Info

X-Velocity: m/s Y-Velocity: m/s Z-Velocity: m/s
 X-Rot. Sp.: rad/s Y-Rot. Sp.: rad/s Z-Rot. Sp.: rad/s

Constraints on forces/moments/motion of body: Advanced

Modify Aero-Force in x: ☐ yes ☒ no Modify Aero-Moment in x: ☐ yes ☒ no
 Modify Aero-Force in y: ☐ yes ☒ no Modify Aero-Moment in y: ☐ yes ☒ no
 Modify Aero-Force in z: ☐ yes ☒ no Modify Aero-Moment in z: ☐ yes ☒ no

Initialization based on: ☒ Euler Angles ☐ Unit Vectors Ignored on 6DOF restart

Initial Euler Angles (degrees):
 Euler_x: Euler_y: Euler_z:

Set Properties For Body 1 Body 1 Definition (BC Selectors) Body 1 External Body Forces

Read 6DOF File Write and Exit Copy Files Cancel

Fig. 11 The 6-DOF specifications

All parameters are with respect to body coordinates, except when defining the initial Euler angles set in “Initialization based on:” select “Euler Angles.” For this example, set the coning angle to 0.5° , to represent the initial body rotation reference from inertial frame. It is important to ensure that the body longitudinal axis is aligned to the global longitudinal axis before setting the Euler angles. Since the rotation is about the global +y-axis, Euler_y is set to 0.5 (note degrees). The body frame and the inertial frame are considered to be initially aligned. If the angle were to be set through a mesh rotation, rather than the Euler Angle, the inertial to body transformation matrix would be incorrect. Therefore, all angles must be set within the 6-DOF module in order for forces and moments to be interpreted correctly in the output.

The “Initial/Current Trans./Rotat. speeds:” prescribes the rotation rates of the body axes so that coning about the inertial x-axis is achieved. The rotation rates are defined with reference to the body coordinate system after the $+0.5^\circ$ Euler_y initialization, therefore to represent the rotation axis that is along the global (inertial) +x-axis, “X-Rot. Sp.:” is set to $\Omega \cdot \cos(\alpha)$, and “Z-Rot. Sp.:” is set to $\Omega \cdot \sin(\alpha)$. This rotational rate is independent of the defined body spin rate (ω) that was defined earlier. In this present example, the rates (Ω and ω) were chosen so that the projectile would counter-rotate, resulting in no projectile rotation about its body x-axis.

To define the boundaries of the body, click “Body 1 Definitions (BC Selectors)” and select each of the boundaries that compose the body (Fig. 12). Finally, click “Write and Exit” to write the 6-DOF input file—mcf6dof.inp. Click “Proceed”. The double-axis of rotation is defined.

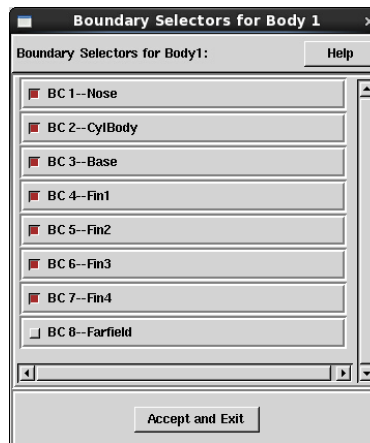


Fig. 12 Definition of body boundaries

In order to ensure convergence, 4 cycles of coning were simulated. The global time step was selected based on the prescribed coning rotation rate. The time step, Δt , is defined as the following:

Approved for public release; distribution is unlimited.

$$\Delta t = \frac{2\pi}{N\dot{\phi}}, \quad (7)$$

where N is the number of numerical time steps per rotation, equivalent to spinning the projectile $\frac{360}{N}$ degree every timestep. For the current investigation, $N = 360, 720,$ and $1,440$ numerical steps were used. It was found that the finer resolved time step ($N = 1,440$) trended toward a more converged solution (i.e., PDM values were approaching an asymptotic value) and resulted in a calculated PDM value closer to the planar pitching calculated PDM value. The total number of global time steps was determined based on the number of global cycles of oscillation desired (4 cycles of coning) and multiplying by the number of steps per cycle (N).

4.3 Kinematic Check

Results of the kinematic motion are shown in Fig. 13. Due to the counter-rotation about the body axis, the projectile's roll angle remains constant while undergoing coning.

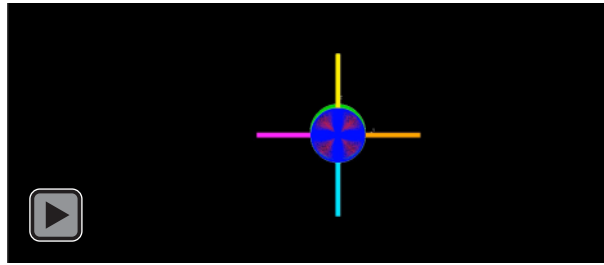


Fig. 13 Kinematic motion of double-axis rotation (coning with counter-rotation body spin) (video)

4.4 Output

The 6-DOF output file (Fig. 14), `mcf6dof_b1.dat`, contains all necessary body forces, body moments, and Euler angles at each time step with respect to the inertial frame. Moreover, the transformation matrix from inertial to body is also provided ("b" matrix). With this information, the body forces and body moments can be determined in the body frame. A detailed description of the 6-DOF output is provided by Metacomp Technologies, Inc.⁸

```

1 TITLE = "mcfd6dof output for Body#1 named Body1"
2 VARIABLES =
3 "move#", "time", "mass", "x-CG", "y-CG", "z-CG",
4 "x-BodyOrig", "y-BodyOrig", "z-BodyOrig", "x-CentRot", "y-CentRot", "z-CentRot",
5 "Ixx", "Iyy", "Izz", "Ixy", "Ixz", "Iyz",
6 "x-AeroForce", "y-AeroForce", "z-AeroForce", "x-AeroMom0", "y-AeroMom0", "z-AeroMom0",
7 "x-AeroMomCR", "y-AeroMomCR", "z-AeroMomCR",
8 "x-TotForce", "y-TotForce", "z-TotForce", "x-TotMomCR", "y-TotMomCR", "z-TotMomCR",
9 "x-Torque", "y-Torque", "z-Torque",
10 "x-LinAcc", "y-LinAcc", "z-LinAcc", "x-AngAcc", "y-AngAcc", "z-AngAcc",
11 "x-LinVel", "y-LinVel", "z-LinVel", "x-AngVel", "y-AngVel", "z-AngVel",
12 "quat0", "quat1", "quat2", "quat3",
13 "a11", "a12", "a13", "b11", "b12", "b13",
14 "a21", "a22", "a23", "b21", "b22", "b23",
15 "a31", "a32", "a33", "b31", "b32", "b33",
16 "x-EulAng", "y-EulAng", "z-EulAng"
17 ZONE F=POINT
18 1 3.0767000e-05 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
19 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
20 1.0000000e+00 1.0000000e+00 1.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
21 1.4120432e-02 -1.5738995e+00 2.3677380e+01 -3.5889641e-03 -1.3297016e+00 -3.6279964e-01
22 -3.5889641e-03 -1.3297016e+00 -3.6279964e-01 0.0000000e+00 0.0000000e+00 0.0000000e+00
23 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
24 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
25 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 8.9105840e+02 7.7761000e+00
26 9.9989653e-01 1.3707559e-02 4.3628993e-03 5.9810138e-05
27 9.9996192e-01 1.5027423e-09 8.7265355e-03 9.9996192e-01 2.3921730e-04 -8.7232561e-03
28 2.3921730e-04 9.9962420e-01 -2.7411759e-02 1.5027423e-09 9.9962420e-01 2.7412803e-02
29 -8.7232561e-03 2.7412803e-02 9.9958614e-01 8.7265355e-03 -2.7411759e-02 9.9958614e-01
30 1.5708945e+00 4.9981210e-01 1.3706663e-02
31 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
32 2 6.1534000e-05 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00
33 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00

```

Fig. 14 Example output of mcfd6dof_b1.dat

The elements of the transformation matrix (from inertial to body) at each time step is provided through the “b” values. From these values, the transformation matrix can be computed. The Euler angles that define the transformation matrix only account for the coning motion, therefore after applying the transformation, the result is in nonrolling body reference frame. An example of the moments in inertial versus nonrolling body frame from the last 2 global cycles for the prescribed coning-spinning motion are shown in Figs. 15–17. The moments in the inertial frame are oscillatory, exhibiting large amplitude with respect to the inertial frame as is expected. After applying the transformation from inertial to nonrolling body frame, the body forces and moments become approximately constant values because the total angular velocity of the body about the longitudinal axis is zero (Figs. 15–17). The transformation to the nonrolling body frame should result in a constant “flat line”; however, the results of the simulation for each time step studied ($\frac{1^\circ}{\Delta t}$, $\frac{0.5^\circ}{\Delta t}$, and $\frac{0.25^\circ}{\Delta t}$ at $\hat{\Omega} = 0.0025$) show an oscillation in the forces and moments. This amplitude appears to be inversely proportional to the defined angular rate of the double-rotation motion. Further analysis should be pursued to determine if the oscillation is a false artifact from the 6-DOF calculating the forces and moments or a result from resolving the transient behavior of the flow physics (i.e., boundary layer and/or wake flow).

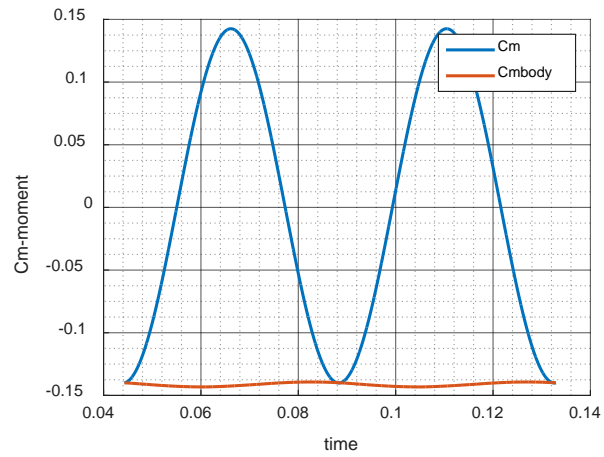


Fig. 15 Cm in inertial and body frame $\left(\frac{0.25^\circ}{\Delta t}\right)$

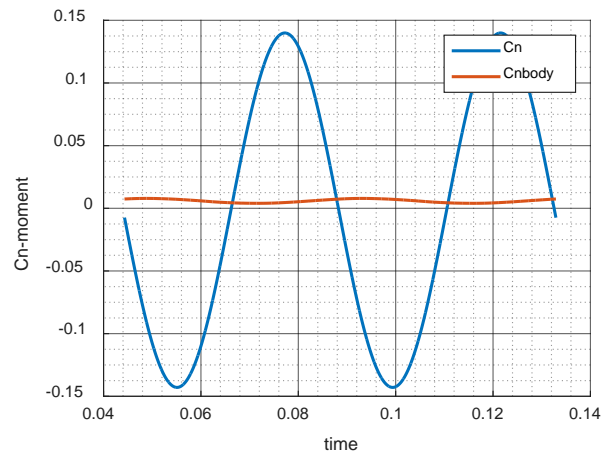


Fig. 16 Cn in inertial and body frame $\left(\frac{0.25^\circ}{\Delta t}\right)$

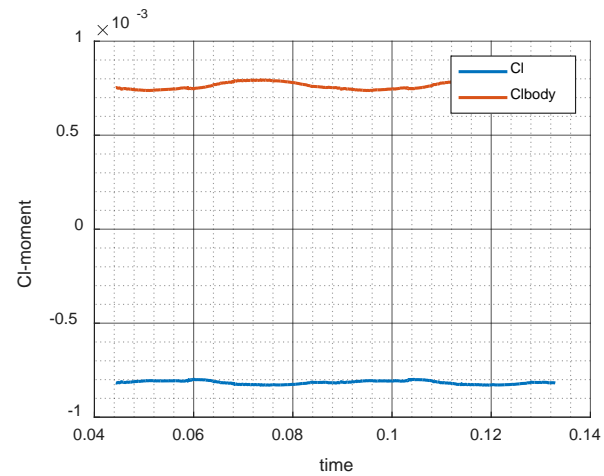


Fig. 17 Cl in inertial and body frame $\left(\frac{0.25^\circ}{\Delta t}\right)$

A negation of the x-moment and the z-moment is completed during the transformation between inertial and body frames so that a direct comparison with previous work can be completed. In this study, a positive x-moment is defined as a counter-clockwise look from the rear of the projectile, while the comparison data are presented where a positive x-moment is defined as clockwise, similarly, with the z-moment.

Once the coefficients are in the nonrolling body ballistic reference frame, calculations for PDM can be performed. The last 2 global cycles of the body forces and moments were averaged (e.g., C_{n_b}) for PDM calculation.

The mcfd.info1 output data file contains the body forces in the inertial frame. Therefore, if one were interested in the body forces on a canard that is attached to the body, one would need to apply the supplied “b” transformation matrix on the data in the mcfd.info1 output data file for the canard to obtain the correct forces in the nonrolling body frame.

5. Results

To ensure that the transformation between the inertial and nonrolling body coordinate systems was being applied correctly in the transient 6-DOF simulation, a transformation matrix was derived analytically (see the Appendix) and compared to the transformation matrix that is outputted from the 6-DOF (i.e., “b” matrix). The derived transformation matrix from inertial (I) to nonrolling coning body frame (B) is presented in Eq. 8.

$$\begin{bmatrix} \hat{i}_B \\ \hat{j}_B \\ \hat{k}_B \end{bmatrix} = \begin{bmatrix} \cos(\sqrt{\theta^2 + \psi^2}) & \sin(\phi) \sin(\sqrt{\theta^2 + \psi^2}) & -\sin(\sqrt{\theta^2 + \psi^2}) \cos(\phi) \\ 0 & \cos(\phi) & \sin(\phi) \\ \sin(\sqrt{\theta^2 + \psi^2}) & -\sin(\phi) \cos(\sqrt{\theta^2 + \psi^2}) & \cos(\phi) \cos(\sqrt{\theta^2 + \psi^2}) \end{bmatrix} \begin{bmatrix} \hat{i}_I \\ \hat{j}_I \\ \hat{k}_I \end{bmatrix}, \quad (8)$$

where ϕ , θ , and ψ are the computed Euler angles (i.e., Euler_x, Euler_y, and Euler_z) at each time step of the simulation. The 2 transformation matrices (derived and computed) were found to be identical, therefore verifying that the computed transformation “b” matrix from the 6-DOF output file correctly transformed the forces and moments into the nonrolling body frame.

To further validate the transformation, the body moments and approximate PDM computed from the steady-state solution of lunar coning was compared to the transient lunar coning method (i.e., double-axis of rotation simulation, with zero-body spin implemented, 0.5° rotation per time step) and is shown in Table 1 for $\hat{\Omega} = 0.0157$. The steady-state solution requires no transformation due to the initial computational set up: a single axis rotation of the projectile’s longitudinal axis at $\alpha = 0.5^\circ$ about the freestream velocity vector was simulated. The values presented

for steady case were the average values over the last 200 iterations; whereas, the values presented for the transient case were averaged over the last 2 global coning cycles.

Table 1 Comparison of steady and transient lunar coning

Coefficient	Steady lunar coning	Transient lunar coning
C_{nb} (Cn_body)	0.0467	0.0420
C_{mb} (Cm_body)	-0.1431	-0.1436
C_{lb} (Cl_body)	0.2629	0.2633
PDM + $C_{np\alpha}$	-340.8229	-306.2509

The differences seen in the results obtained from the steady-state and transient simulations is likely attributed to using too large of a time step for the transient simulation resulting in an incomplete convergence of simulation. The size of the time step is addressed once body spin is added, and is explained further below. Regardless, the moments calculated from each are similar in magnitude once the transient coefficients are averaged indicating that the correct transformation is being applied.

To further illustrate the transformation, the forces and moments of the transient lunar coning motion in inertial and nonrolling body frame is presented in Fig. 18. The body forces and moments resulted in approximately constant values with respect to time, as for this case (zero-body spinning) the nonrolling body frame was exactly aligned to the body-fixed frame (orange line). These results suggest that the computed transformation matrix correctly transforms the inertial forces and moments into the nonrolling body frame.

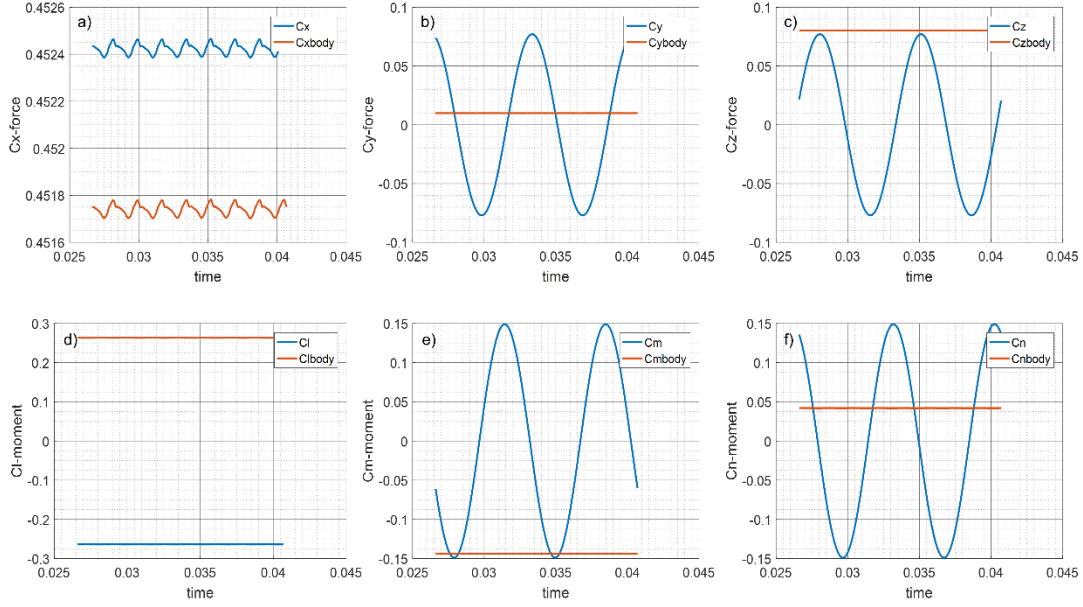


Fig. 18 Transformation of forces and moments from inertial to body frame for transient lunar coning motion $\left(\frac{0.5^\circ}{\Delta t}\right)$

The PDM stability derivative can be compared between transient planar pitching and transient coning-spinning. Only transient planar pitching and transient coning-spinning can calculate the PDM without any additional information from another simulation. The transient lunar coning method does not account for the Magnus moment produced by the body spinning motion (i.e., $C_{n_{p\alpha}}$), therefore resulting into an inaccurate PDM magnitude.³

The results of the double-axis (coning and spinning) method with $\hat{\Omega} = 0.0025$ at different angular resolutions are compared to those obtained from the planar pitching method in Table 2.

Table 2 Comparison of transient methods computing PDM

Coefficient	Transient planar pitching	Transient coning-spinning $\left(\frac{1^\circ}{\Delta t}\right)$	Transient coning-spinning $\left(\frac{0.5^\circ}{\Delta t}\right)$	Transient coning-spinning $\left(\frac{0.25^\circ}{\Delta t}\right)$
C_{n_b} (Cn_body)	NA	0.0038	0.0052	0.0059
C_{m_b} (Cm_body)	-0.1415	-0.1414	-0.1413	-0.1413
C_{l_b} (Cl_body)	NA	4.1949e-04	0.0013	7.6412e-04
PDM	-287.6095	-172.6798	-238.7294	-269.8829

As the resolution of the rotation of the body per time step increases (i.e., time step decreases), the calculated PDM appears to be converging to the transient planar pitching PDM value. Therefore, the results suggest that the body rotation must be fully resolved (i.e., global time step must be small enough) for convergence to be achieved.

Table 3 compares the computed moments and PDM to the results from Bhagwandin and Sahu.³

Table 3 Comparison of computed PDM

$M = 2.5, \alpha = 0.5^\circ,$ $\bar{\Omega} = 0.0025, k = 0.05,$ $A = 0.25^\circ$	Bhagwandin and Sahu (2014) “Truth”	Transient planar pitching	Transient coning-spinning 0.25° Δt
C_{n_b} (Cn_body)	0.0063	N/A	0.0059
C_{m_b} (Cm_body)	-0.1415	-0.1415	-0.1413
C_{l_b} (Cl_body)	0.0	N/A	0.0007
PDM	-289.66	-287.6095	-269.8829

The predicted moments and PDM compare quite well to the work performed by Bhagwandin and Sahu.³ The results are within 6.8%, which is reasonable when compared to ordinary experimental PDM errors that are on the order of at least 15%. The difference in PDM computed by transient planar pitching may be attributed to the planar pitching motion for the current investigation was about $\alpha_0 = 0.25^\circ$, whereas $\alpha_0 = 0^\circ$ in Bhagwandin and Sahu.³ The difference in PDM computed by the double-axis rotation can be attributed to the difference in the computation of C_{n_b} . Bhagwandin and Sahu³ computed the resulting C_{n_b} by subtracting the Magnus moment computed in a separate transient roll simulation from the steady lunar coning computed moment; whereas, the current double-axis rotation simulation inherently removes the Magnus component through the counter spinning motion of the body.

6. Conclusions

A detailed implementation of double-axis rotation in CFD++ was presented. Moreover, an investigation was performed to verify the implementation by computing the PDM stability derivative as described by Weinacht et al.¹ The calculated moments from the double-axis rotation was compared to the computed values from the detailed work of Bhagwandin and Sahu.³ Although the PDM from the coning-spinning simulation was not identical to that determined by Bhagwandin and Sahu,³ the results were reasonable. Reducing the global time step of the transient coning-spinning simulation resulted in a closer match, suggesting that

resolution of the motion must be sufficient in order to achieve convergence. The double-axis spinning-coning methodology presented here is not being proposed as the preferred method to calculate PDM as the planar pitching is likely the most accurate and efficient. This study was completed only as a confirmation of the methodology should it be needed to simulate more complex spinning/coning vehicle motions in the future. The results presented here provide assurance that these complex motions can be accurately simulated using CFD++ and that the forces and moments are properly analyzed from the resulting output.

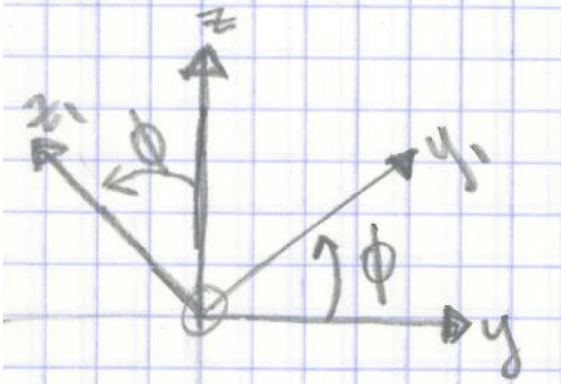
7. References

1. Weinacht P, Sturek WB, Schiff LB. Navier-Stokes predictions of pitch damping for axisymmetric projectiles. *Journal of Spacecraft and Rockets*.1997;34(6).
2. Metacomp Technologies Inc. CFD++ User Manual. Agoura Hills (CA): Metacomp Technologies Inc.; 2014.
3. Bhagwandin VA, Sahu J. Numerical prediction of pitch damping stability derivatives for finned projectiles. *Journal of Spacecraft and Rockets*. 2014;51(5).
4. Dupuis A. Aeroballistic range and wind tunnel tests of the basic finner reference projectile from subsonic to high supersonic velocities. Defense Research and Development, Canada: Valcartier (QC); 2002. TM-2002-136.
5. DeSpirito J, Siltan S, Weinacht P. Navier-Stokes predictions of dynamic stability derivatives: evaluation of steady-state methods. *Journal of Spacecraft and Rockets*. 2009;46(6).
6. Weinacht P, Sturek WB. Navier-Stokes predictions of pitch damping for finned projectiles using steady coning motion. AIAA Paper 90-3088. *Proceedings of the AIAA 8th Applied Aerodynamics Conference*. AIAA. Washington (DC): 1990. p. 632-642.
7. Metacomp Technologies Inc. Double-axis rotational body motion in CFD++. Agoura Hills (CA): Metacomp Technologies Inc.; 2011.
8. Metacomp Technologies Inc. 6DOF numerical data output. Agoura Hills (CA): Metacomp Technologies Inc.; 2011.

Appendix. Transformation from Inertial (I) to Nonrolling Coning Frame (B)

This appendix appears in its original form, without editorial change.
Approved for public release; distribution is unlimited.

1. First Transformation

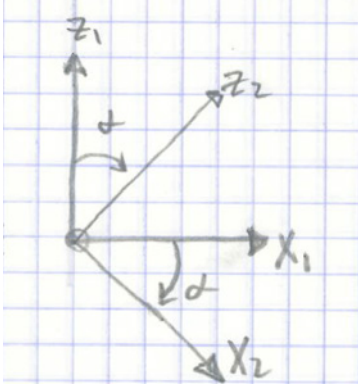


$$\begin{aligned}\hat{i}_1 &= \hat{i}_I \\ \hat{j}_1 &= \hat{j}_I \cos(\phi) + \hat{k}_I \sin(\phi) \\ \hat{k}_1 &= -\hat{j}_I \sin(\phi) + \hat{k}_I \cos(\phi)\end{aligned}$$

$$\therefore \begin{bmatrix} \hat{i}_1 \\ \hat{j}_1 \\ \hat{k}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \hat{i}_I \\ \hat{j}_I \\ \hat{k}_I \end{bmatrix}$$

2. Second Transformation

$$\alpha = \sqrt{\theta^2 + \psi^2}$$



\therefore

$$\begin{aligned}\hat{i}_B &= \hat{i}_1 \cos(\alpha) - \hat{k}_1 \sin(\alpha) \\ \hat{j}_B &= \hat{j}_1 \\ \hat{k}_B &= \hat{i}_1 \sin(\alpha) + \hat{k}_1 \cos(\alpha)\end{aligned}$$

$$\therefore \begin{bmatrix} \hat{i}_B \\ \hat{j}_B \\ \hat{k}_B \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \hat{i}_1 \\ \hat{j}_1 \\ \hat{k}_1 \end{bmatrix}$$

3. Inertial Frame to Body Frame

$$\begin{bmatrix} \hat{i}_B \\ \hat{j}_B \\ \hat{k}_B \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \hat{i}_I \\ \hat{j}_I \\ \hat{k}_I \end{bmatrix}$$

$$\therefore \begin{bmatrix} \hat{i}_B \\ \hat{j}_B \\ \hat{k}_B \end{bmatrix} = \begin{bmatrix} \cos(\sqrt{\theta^2 + \psi^2}) & \sin(\phi) \sin(\sqrt{\theta^2 + \psi^2}) & -\sin(\sqrt{\theta^2 + \psi^2}) \cos(\phi) \\ 0 & \cos(\phi) & \sin(\phi) \\ \sin(\sqrt{\theta^2 + \psi^2}) & -\sin(\phi) \cos(\sqrt{\theta^2 + \psi^2}) & \cos(\phi) \cos(\sqrt{\theta^2 + \psi^2}) \end{bmatrix} \begin{bmatrix} \hat{i}_I \\ \hat{j}_I \\ \hat{k}_I \end{bmatrix}$$

List of Symbols, Abbreviations, and Acronyms

ANF	Army-Navy Finner
ARL	US Army Research Laboratory
CFD	computational fluid dynamics
CFL	Courant-Friedrichs-Lewy
DOF	degree of freedom
DSRC	Department of Defense Supercomputing Resource Center
PDM	pitch-damping moment
RANS	Reynolds Averaged Navier-Stokes

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA	RDRL WML E J VASILE V BHAGWANDIN I CELMINS J DESPIRITO S SILTON F FRESCONI J SAHU P WEINACHT RDRL WML F T BROWN M ILG B KLINE J MALEY B NELSON RDRL WML G M MINNICINO J SOUTH RDRL WML H J NEWILL RDRL WMP D LYON RDRL WMS H E MAUPIN
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO L IMAL HRA MAIL & RECORDS MGMT	
1 (PDF)	GOVT PRINTG OFC A MALHOTRA	
5 (PDF)	AMRDEC L AUMAN J DOYLE K KENNEDY M MCDANIEL C ROSEMA	
6 (PDF)	ARDEC M STOLK YIN CHEN T RECCHIA D HOSIER M DUCA G RODEBAUGH	
1 (PDF)	NWSCDD G33 S KOSKI	
1 (PDF)	USMA DEPT OF MECHANICAL ENGINEERING C VERHULST	
28 (PDF)	DIR USARL RDRL WM B FORCH S KARNA J LA SCALA S SCHOENFELD J ZABINSKI RDRLWML W OBERLE P PEREGINO RDRL WML B N TRIVEDI RDRL WML C S AUBERT RDRL WML D R BEYER	